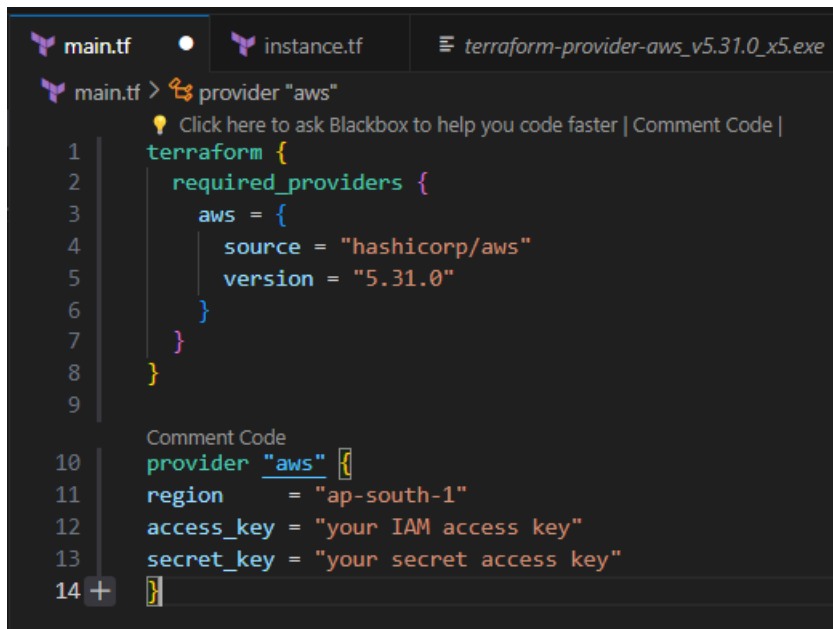


EXPERIMENT – 3

Name: - Shashwat. Dnyaneshwar Kamdi
Batch – 2 [DevOps Non-Hons]
SAP ID- 500092140
Subject – System Provisioning and Configuration Management Lab

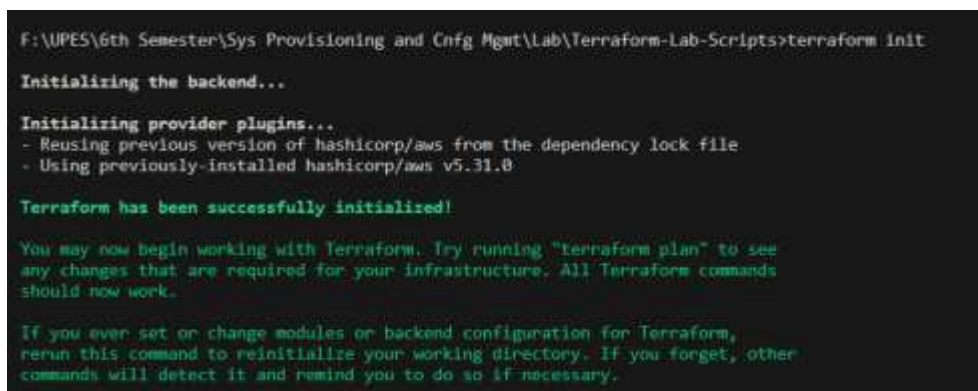
Aim: Provisioning an EC2 Instance on AWS.

1] Create a Terraform Configuration File (main.tf)



```
main.tf | instance.tf | terraform-provider-aws_v5.31.0_x5.exe
main.tf > provider "aws"
  Click here to ask Blackbox to help you code faster | Comment Code |
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.31.0"
6     }
7   }
8 }
9
10 Comment Code
11 provider "aws" {
12   region = "ap-south-1"
13   access_key = "your IAM access key"
14   secret_key = "your secret access key"
15 }
```

2] Initialize Terraform.



```
F:\UPES\6th Semester\Sys Provisioning and Cnfg Mgmt\Lab\Terraform-Lab-Scripts>terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.31.0

Terraform has been successfully initialized!

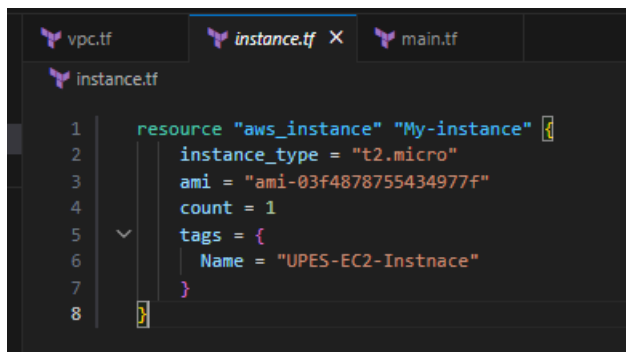
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

3] Validate the Script.

```
F:\UPES\6th Semester\Sys Provisioning and Cnfg Mgmt\Lab\Terraform-Lab-Scripts>terraform validate
Success! The configuration is valid.
```

4] Create a Terraform Configuration File for EC2 instance (instance.tf).



```
vpc.tf  instance.tf X  main.tf
instance.tf
1  resource "aws_instance" "My-instance"
2      instance_type = "t2.micro"
3      ami = "ami-03f4878755434977f"
4      count = 1
5      tags = {
6          Name = "UPES-EC2-Instnace"
7      }
8  }
```

5] Review Plan using Command “Terraform plan”



```
F:\UPES\6th Semester\Sys Provisioning and Cnfg Mgmt\Lab\Terraform-Lab-Scripts>terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.My-instance[0] will be created
+ resource "aws_instance" "My-instance" {
  + ami                         = "ami-03f4878755434977f"
  + arn                        = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone           = (known after apply)
  + cpu_core_count              = (known after apply)
  + cpu_threads_per_core        = (known after apply)
  + disable_api_stop            = (known after apply)
  + disable_api_termination     = (known after apply)
  + ebs_optimized               = (known after apply)
  + get_password_data           = false
  + host_id                     = (known after apply)
  + host_resource_group_arn     = (known after apply)
  + iam_instance_profile        = (known after apply)
  + id                          = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle          = (known after apply)
  + instance_state              = (known after apply)
  + instance_type               = "t2.micro"
  + ipv6_address_count          = (known after apply)
  + ipv6_addresses              = (known after apply)
  + key_name                     = (known after apply)
  + monitoring                  = (known after apply)
  + outpost_arn                 = (known after apply)
  + password_data               = (known after apply)
  + placement_group             = (known after apply)
  + placement_partition_number  = (known after apply)

+ instance_lifecycle           = (known after apply)
+ instance_state               = (known after apply)
+ instance_type                = "t2.micro"
+ ipv6_address_count           = (known after apply)
+ ipv6_addresses               = (known after apply)
+ key_name                     = (known after apply)
+ monitoring                   = (known after apply)
+ outpost_arn                  = (known after apply)
+ password_data                = (known after apply)
+ placement_group              = (known after apply)
+ placement_partition_number    = (known after apply)
+ primary_network_interface_id = (known after apply)
+ private_dns                  = (known after apply)
+ private_ip                   = (known after apply)
+ public_dns                   = (known after apply)
+ public_ip                    = (known after apply)
+ secondary_private_ips        = (known after apply)
+ security_groups               = (known after apply)
+ source_dest_check            = true
+ spot_instance_request_id     = (known after apply)
+ subnet_id                    = (known after apply)
+ tags                         = {
  + "Name" = "UPES-EC2-Instnace"
}
+ tags_all                     = {
  + "Name" = "UPES-EC2-Instnace"
}
+ tenancy                      = (known after apply)
+ user_data                    = (known after apply)
+ user_data_base64             = (known after apply)
+ user_data_replace_on_change  = false
+ vpc_security_group_ids       = (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
```

6] Apply it using command “Terraform apply”

```
F:\UPES\6th Semester\Sys Provisioning and Cnfg Mgmt\Lab\Terraform-Lab-Scripts>terraform apply
```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

```
# aws_instance.My-instance[0] will be created
+ resource "aws_instance" "My-Instance" {
  + ami                    = "ami-03f48787554349774"
  + ami                   = (known after apply)
  + arn                   = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone      = (known after apply)
  + cpu_core_count        = (known after apply)
  + cpu_threads_per_core  = (known after apply)
  + disable_api_stop      = (known after apply)
  + disable_api_termination = (known after apply)
  + ebs_optimized         = (known after apply)
  + get_password_data     = false
  + host_id               = (known after apply)
  + host_resource_group_arn = (known after apply)
  + iam_instance_profile   = (known after apply)
  + id                   = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle    = (known after apply)
  + instance_state        = (known after apply)
  + instance_type         = "t2.micro"
  + ipv6_address_count    = (known after apply)
  + ipv6_addresses       = (known after apply)
  + key_name              = (known after apply)
  + monitoring            = (known after apply)
  + outpost_arn           = (known after apply)
  + password_data         = (known after apply)
  + placement_group       = (known after apply)
  + placement_partition_number = (known after apply)
  + primary_network_interface_id = (known after apply)
  + private_dns           = (known after apply)
  + private_ip            = (known after apply)
```

```
+ subnet_id              = (known after apply)
+ tags                   = {
  + "Name" = "UPES-EC2-Instnace"
}
+ tags_all               = {
  + "Name" = "UPES-EC2-Instnace"
}
+ tenancy                 = (known after apply)
+ user_data               = (known after apply)
+ user_data_base64       = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}
```

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?

Terraform will perform the actions described above.

Only 'yes' will be accepted to approve.

Enter a value: yes

aws_instance.My-instance[0]: Creating...

aws_instance.My-instance[0]: Still creating... [10s elapsed]

aws_instance.My-instance[0]: Still creating... [20s elapsed]

aws_instance.My-instance[0]: Still creating... [30s elapsed]

aws_instance.My-instance[0]: Creation complete after 33s [id=i-0de64cd70bf7cc382]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

7] Verify Resources on AWS Management Console.

Instances (1) Info								
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/>								
<input type="button" value="Instance state = running"/> <input type="button" value="Clear filters"/>								
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv
<input type="checkbox"/>	UPES-EC2-Inst...	i-0de64cd70bf7cc382	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1a	ec2-13-23

Instance summary for i-0de64cd70bf7cc382 (UPES-EC2-Instance) [info](#)

Updated less than a minute ago

Instance ID i-0de64cd70bf7cc382 (UPES-EC2-Instance)	Public IPv4 address 13.232.61.4 open address	Private IPv4 addresses 172.31.46.86
IPv6 address --	Instance state Running	Public IPv4 DNS ec2-13-232-61-4.ap-south-1.compute.amazonaws.com open address
Hostname type IP name: ip-172-31-46-86.ap-south-1.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-46-86.ap-south-1.compute.internal	
Answer private resource DNS name --	Instance type t2.micro	Elastic IP addresses --
Auto-assigned IP address 13.232.61.4 [Public IP]	VPC ID vpc-0ee0942542c3f2cc open	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
IAM Role --	Subnet ID subnet-09ef7cc612c8ddf5c open	Auto Scaling Group name --
IMDSv2 Optional		

8] Cleanup Resources using command “Terraform destroy”

```
F:\UPES\6th Semester\Sys Provisioning and Cnfg Mgmt\Lab\Terraform-Lab-Scripts>terraform destroy
aws_instance.My-instance[0]: Refreshing state... [id=i-0de64cd70bf7cc382]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
- destroy

Terraform will perform the following actions:

# aws_instance.My-instance[0] will be destroyed
- resource "aws_instance" "My-instance" {
  ami              = "ami-03f48767554349774" -> null
  arn              = "arn:aws:ec2:ap-south-1:394464388823:instance/i-0de64cd70bf7cc382" -> null
  associate_public_ip_address = true -> null
  availability_zone = "ap-south-1a" -> null
  cpu_core_count    = 1 -> null
  cpu_threads_per_core = 1 -> null
  disable_api_stop   = false -> null
  disable_api_termination = false -> null
  ebs_optimized      = false -> null
  get_password_data  = false -> null
  hibernation        = false -> null
  id                = "i-0de64cd70bf7cc382" -> null
  instance_initiated_shutdown_behavior = "stop" -> null
  instance_state     = "running" -> null
  instance_type      = "t2.micro" -> null
  ipv6_address_count = 0 -> null
  ipv6_addresses     = [] -> null
  monitoring         = false -> null
}
```

Plan: 0 to add, 0 to change, 1 to destroy.

Do you really want to destroy all resources?

Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

```
aws_instance.My-instance[0]: Destroying... [id=i-0de64cd70bf7cc382]
aws_instance.My-instance[0]: Still destroying... [id=i-0de64cd70bf7cc382, 10s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-0de64cd70bf7cc382, 20s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-0de64cd70bf7cc382, 30s elapsed]
aws_instance.My-instance[0]: Destruction complete after 30s
```

Destroy complete! Resources: 1 destroyed.

Instances [info](#)

Instance state = running [X](#) [Clear filters](#)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
No instances						
You do not have any instances in this region						
Launch instances						